2021

Boilin' Quick Cooking Timer

DEVELOPER GUIDE

MADE BY: CADEN FRIESEN

System Overview

This system is designed with two main functions in mind. One function is the ability to keep track of and display a timer for up to 99 minutes. The system will set off an alarm when the time hits zero and the time can be set to any increment of 1 minute. The second feature is a boiling detector probe that can be put in a pot of water and when turned on will set off the alarm if the water reaches 98.5 degrees Celsius which is the typical rolling boil point. The system also comes with three brightness settings and flexibility for starting and resetting the timer any time. It is powered by a typical 5-volt adapter wall plug connected to the Arduino Nano inside the case.

Electrical Specifications

These electrical specifications are based off the requirements of the Arduino Nano that drives the system.

Table 1: Electrical Limitations and Maximums for Operation

Operating Voltage	5 Volts
Plug in Max Voltage	240 Volts
Plug in Min Voltage	100 Volts
Operating Current	1 Ampere
Min/Max Temperature	-40 C to 85 C

User Guide

--Powering the Device--
To turn on the device the Mini-B USB must be plugged in to the back of the box. The USB comes with a 5-volt wall adaptor that can be plugged into any household 100-240 volt wall outlet. Plugging this in will power the system in its starting state. The system will start at 0 seconds with the boiling feature off and the time not counting down. At this point the features laid out below can be used.

--Detaching the boiling detector probe--
The boiling detector probe can be disconnected for easy transport by pinching the top and bottom of the white connector attaching it to the system. This can easily be clipped back on by connecting the two clasps back together on the bottom of the connector. The system can run without this probe plugged in, but the boiling feature will not be usable.

--Boiling Feature--
The boiling feature on this device can be used to detect a rolling boil in a water source and alert the user via the 440 Hz alarm. To set up this feature use the probe coming out of the right side of the device. Place the gel tip of the probe so that it is submerged in the water source being heated. The food safe heat covering can be placed on the side of a pan and will slightly bend to help this process. Do not submerge the timer box or any part of the probe past the food safe

heat covering to prevent damage to the unit. Once the probe is in the water turn on the boiling feature using the BOIL button and the alarm will activate once the probe has reached 98 degrees Celsius. The alarm can be deactivated by pressing the BOIL button again or removing the probe from the water.

--Buttons--
This system contains 6 buttons for the user interface with labels beneath each button for identification. These six buttons will be explained below starting from left to right on the top row then continuing to the bottom row.

--BRT – Yellow--
This button is used to control the brightness level of the timer display. The Display will start on the high setting when the system is turned on. Pressing this button once will decrease the brightness to the medium setting. Pressing it again will decrease the brightness to the low setting. The next press will loop the brightness back to the high setting.

--1 MIN – Black--
This button will add one minute to the timer. It can be pressed while the timer is stopped or while it is running without interfering with the current counting.

--5 MIN – Blue--
This button will add five minutes to the timer. It can be pressed while the timer is stopped or while it is running without interfering with the current counting.

--BOIL – White--
This button will turn on sampling for the boiling feature. The boiling feature will not record samples or set off the alarm if the LED on the right of the box is not on. Pressing this button will activate this sampling and turn on the LED. Pressing the button again will turn off sampling and turn off the LED.

--START – Green--
This button starts the timer in the system. In its initial state the timer will not count down. Once the green button is pressed the timer will begin counting down and will trigger the alarm once it hits zero. Time can be added after the green button is pressed and will deactivate the alarm noise if the timer is at zero when the time is added. The alarm will reactivate when the timer reaches zero again.

Block Diagrams

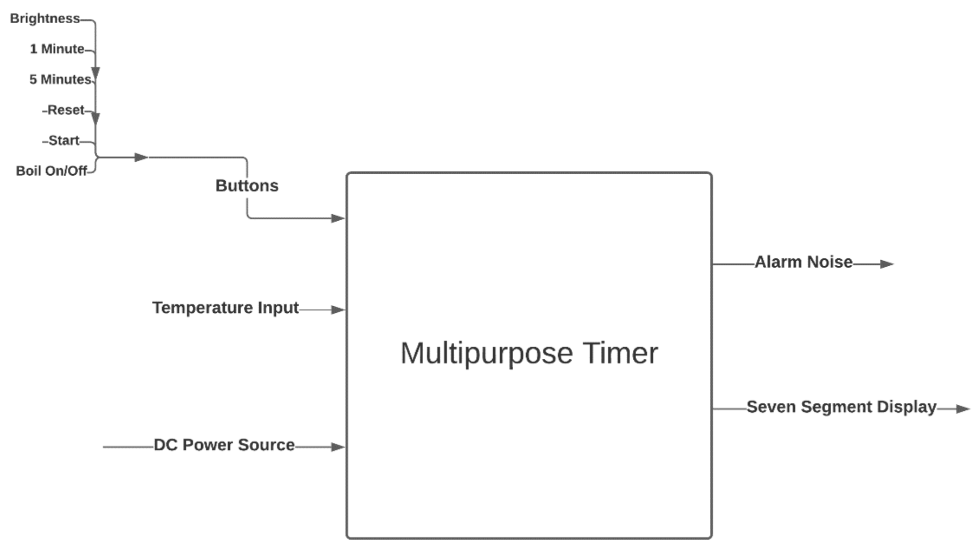


Figure 1: Black Box Diagram

This block diagram illustrates all the external inputs and outputs to the total overall system.

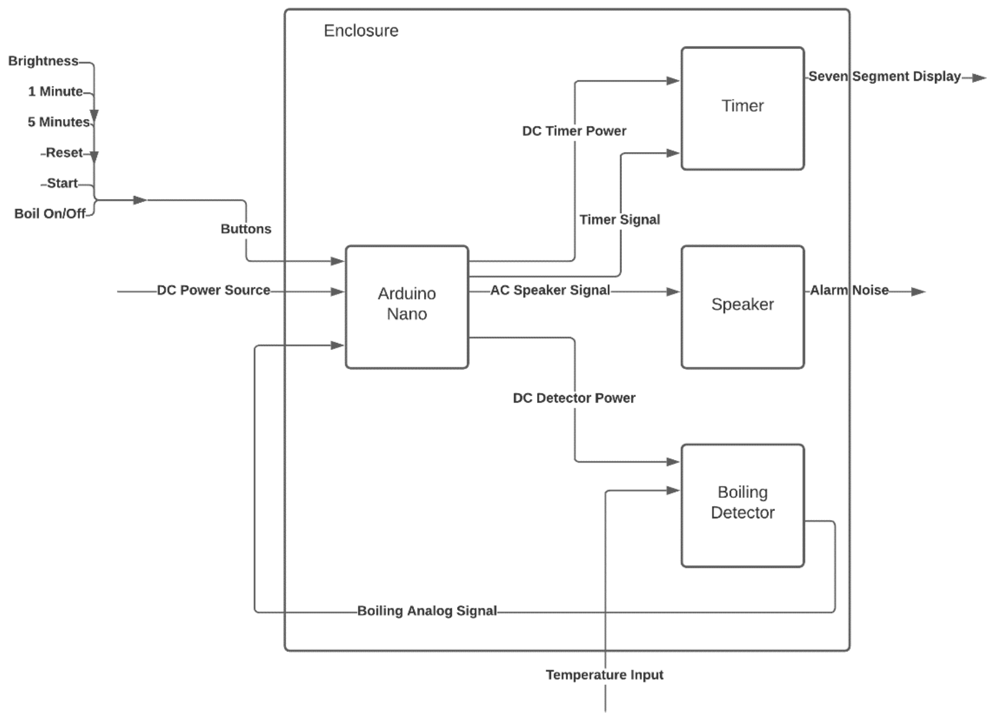


Figure 2: Top Level Block Diagram

This illustrates all the interfaces used to communicate between blocks inside and outside the system. It also shows what interfaces are contained within the enclosure and what is not.

Electrical Schematics

This is the schematic of the entire system. All red dots represent connections and wires passing over each other without these dots are not connected. It is also important to note that the arduino nano will be plugged into the power cord when the system is running.

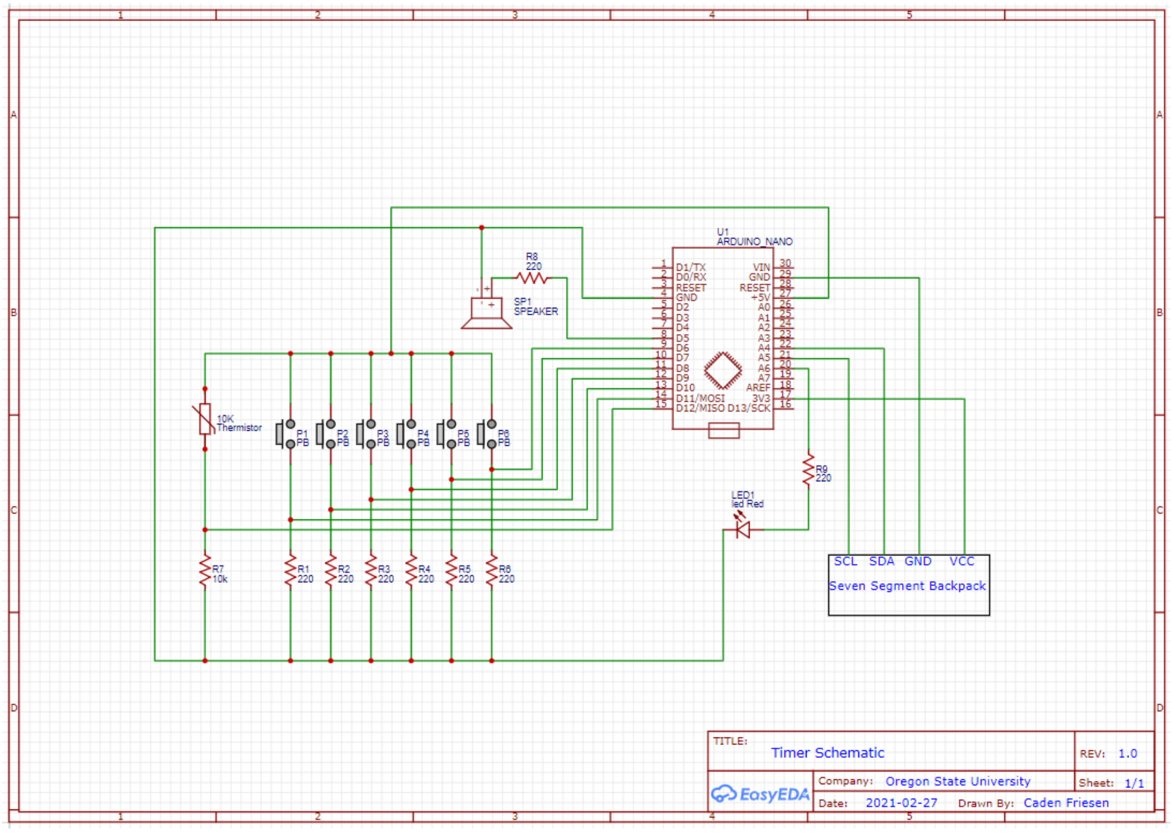


Figure 3: Electrical Schematic

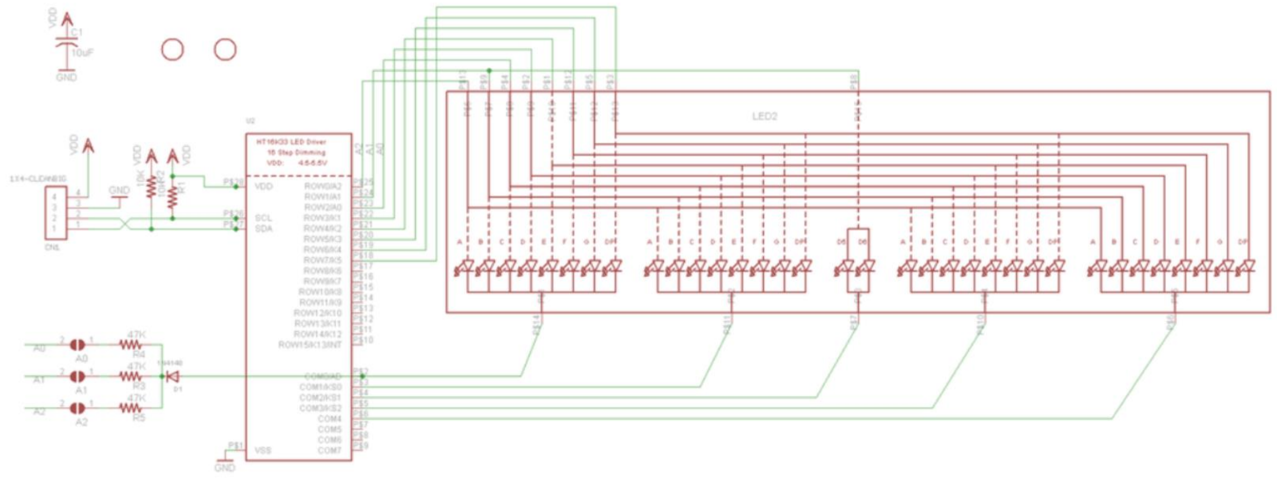


Figure 4: Seven Segment Backpack Schematic (Provided by Adafruit)

Mechanical Drawings

Below mechanical drawings of all the 3d printed parts and internal devices has been included. These were made in solidworks and all measurements are in millimeters unless specified otherwise.

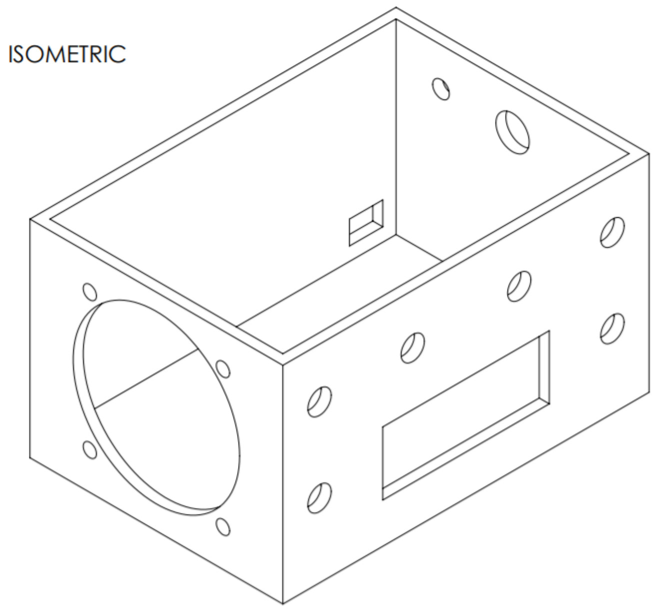


Figure 5: Isometric view of the 3d printed enclosure

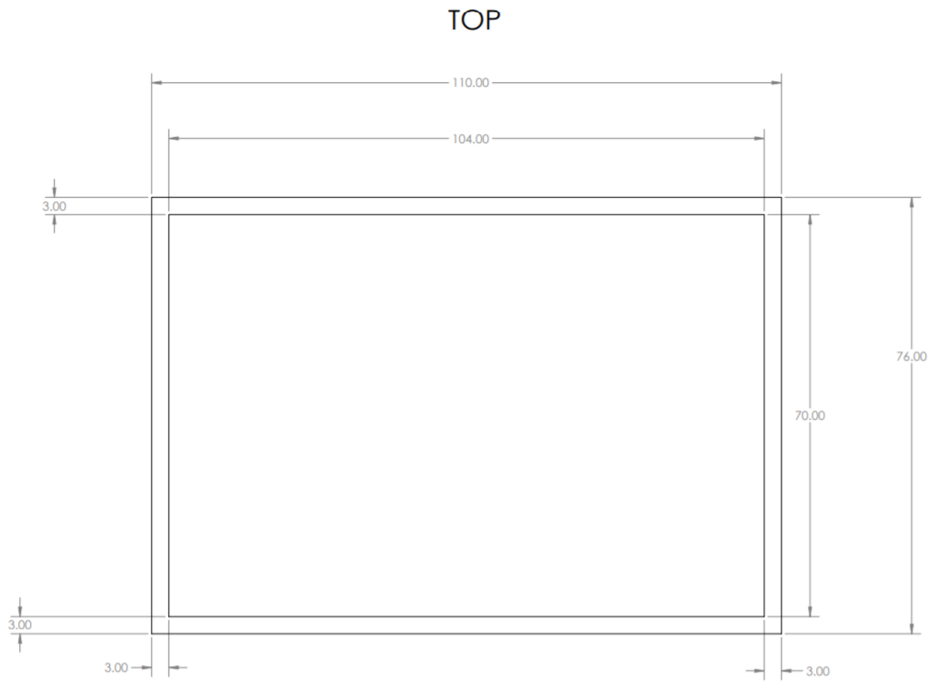


Figure 6: Top view of the enclosure

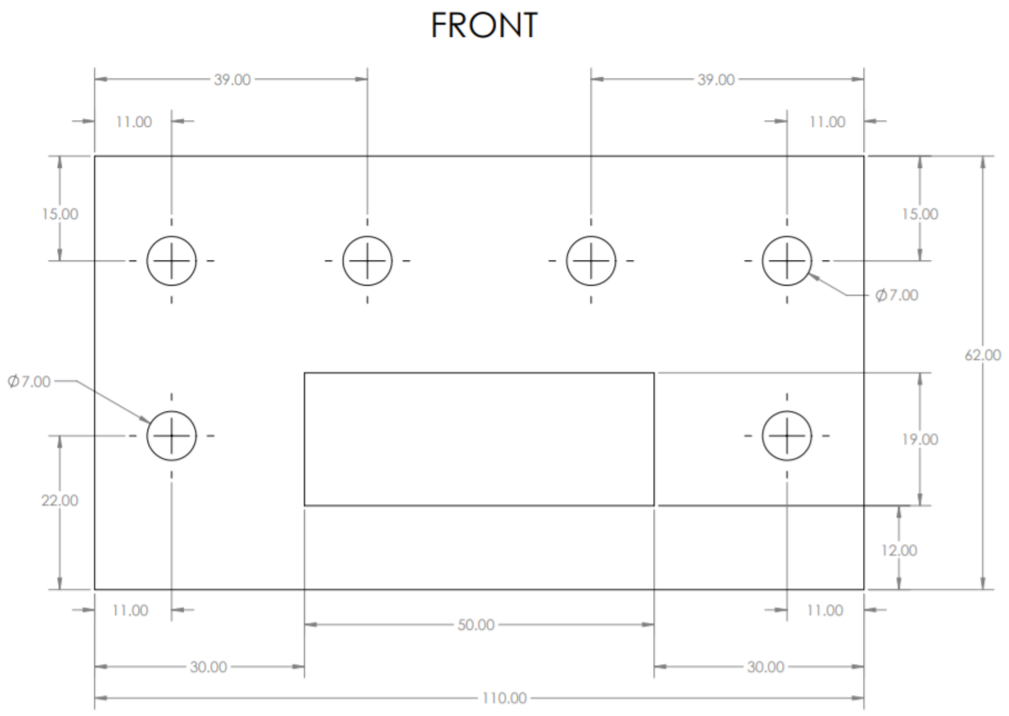


Figure 7: Front of the enclosure

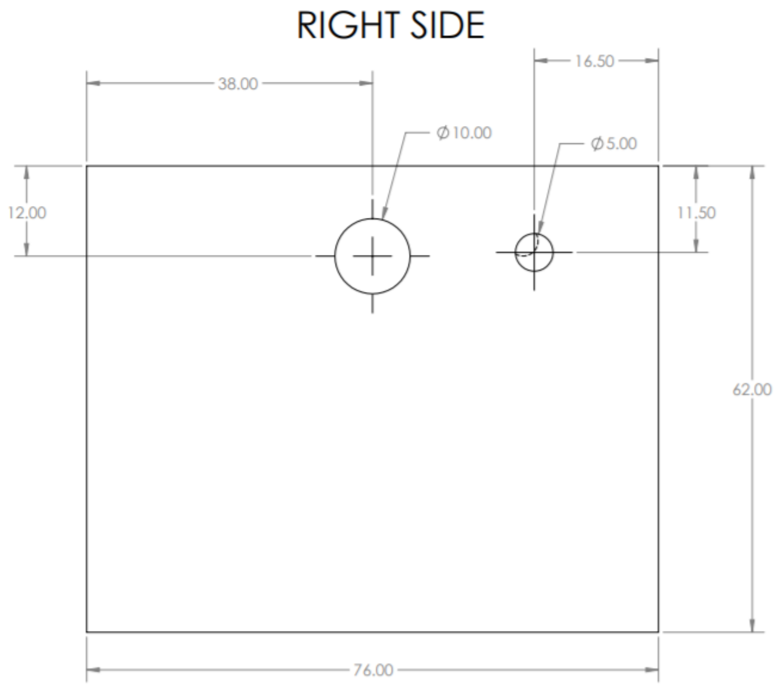


Figure 8: Right side of the enclosure

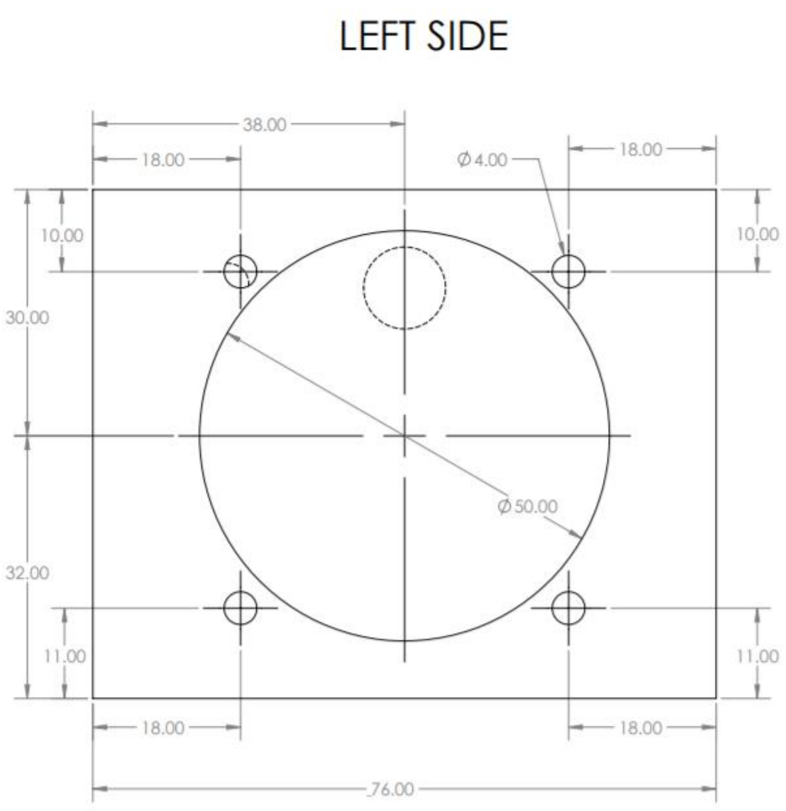


Figure 9: Left side of the enclosure

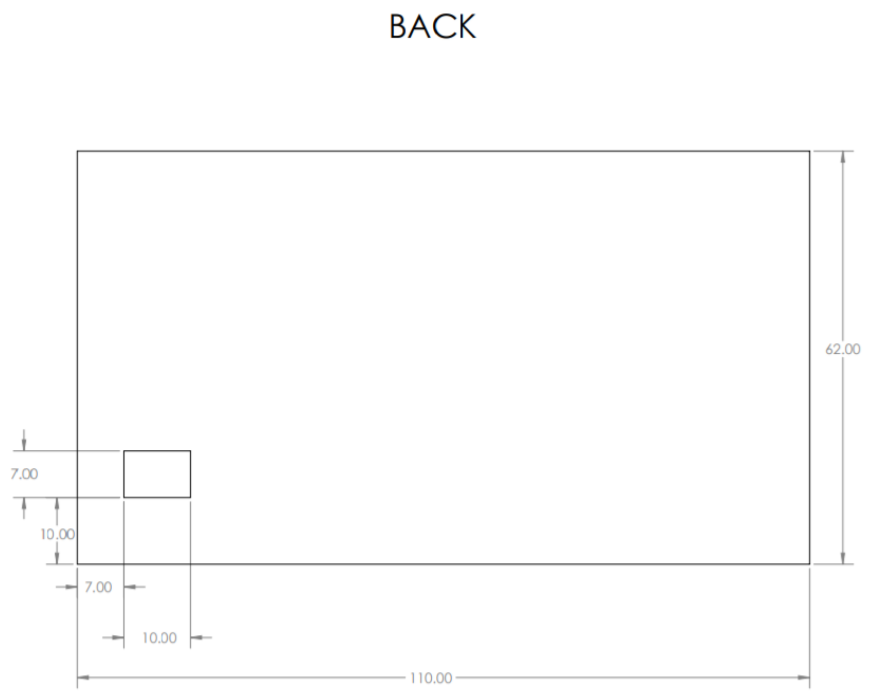


Figure 10: Back side of the enclosure

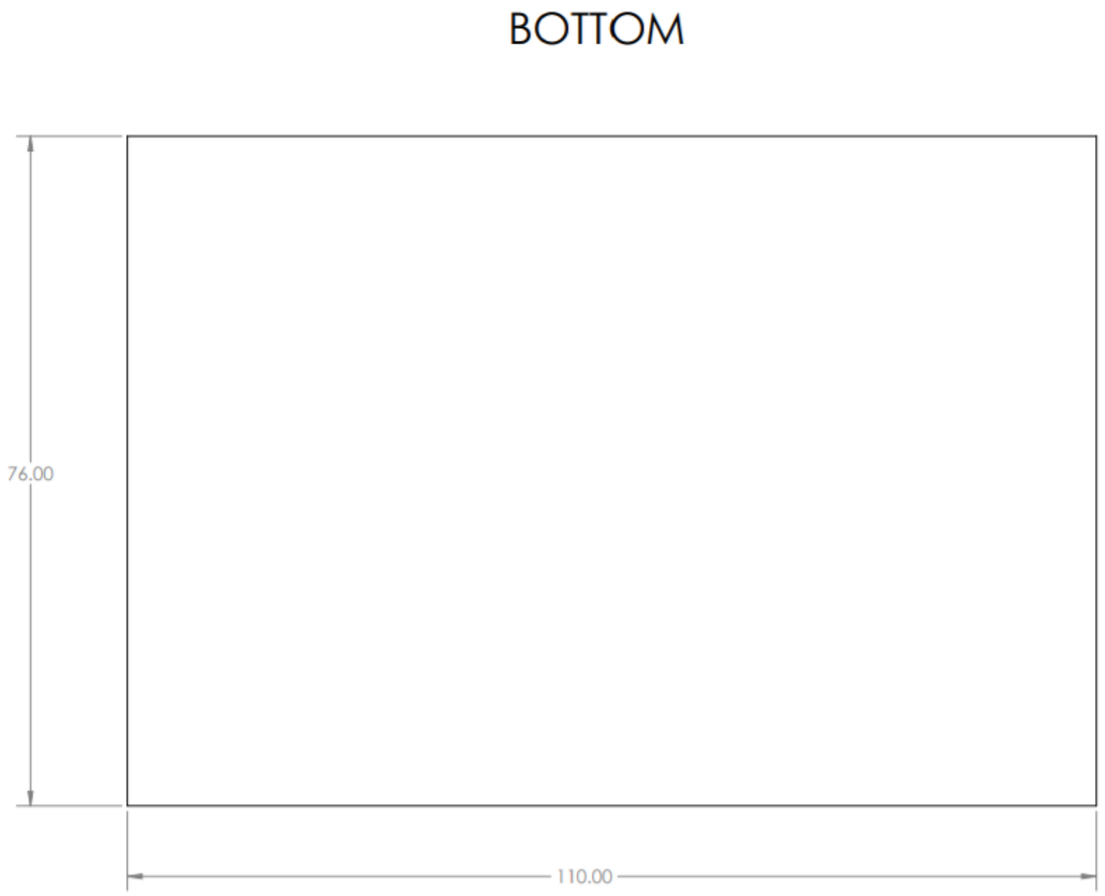


Figure 11: Bottom of the enclosure (Solid)

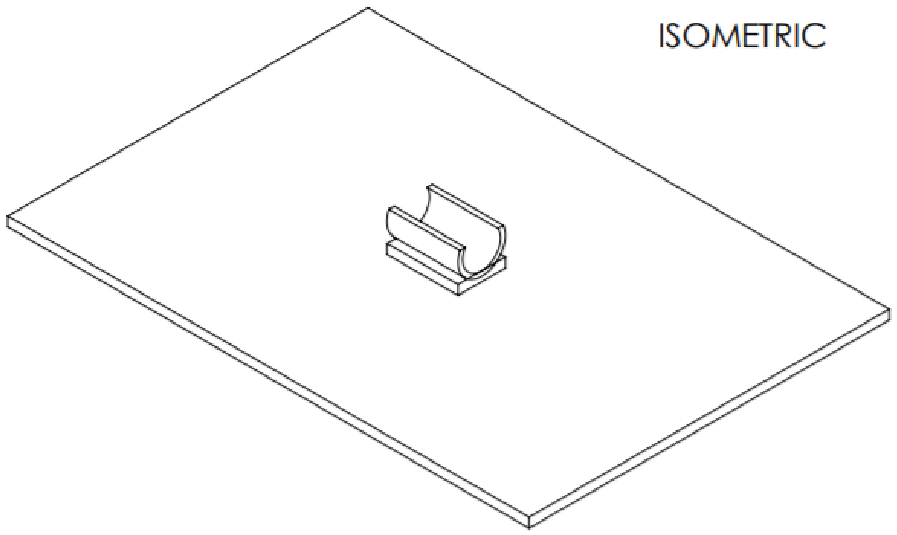


Figure 12: Isometric view of lid of enclosure

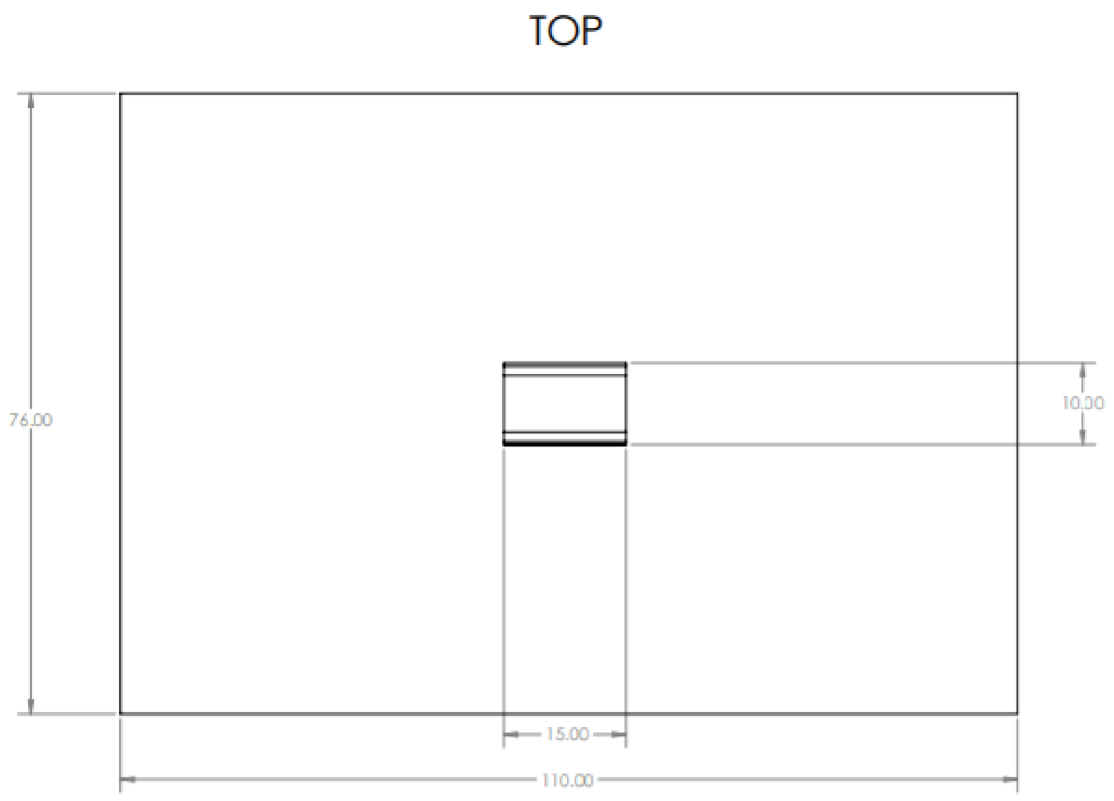


Figure 13: Top view of enclosure lid

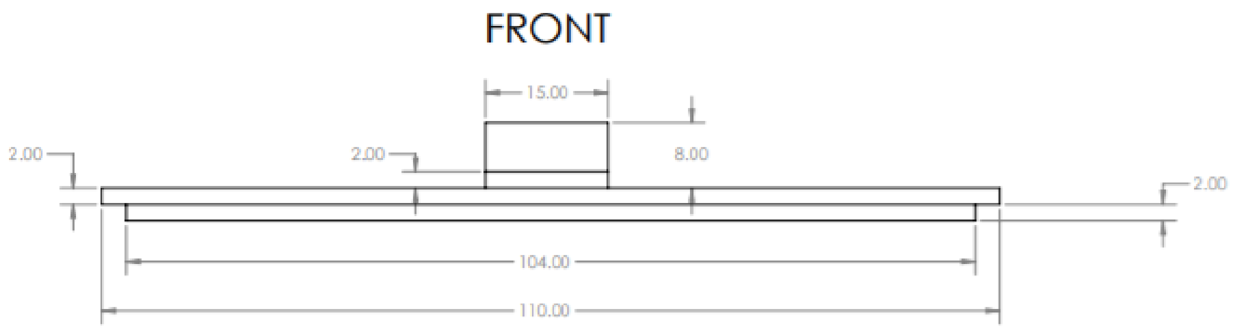


Figure 14: Front view of enclosure lid

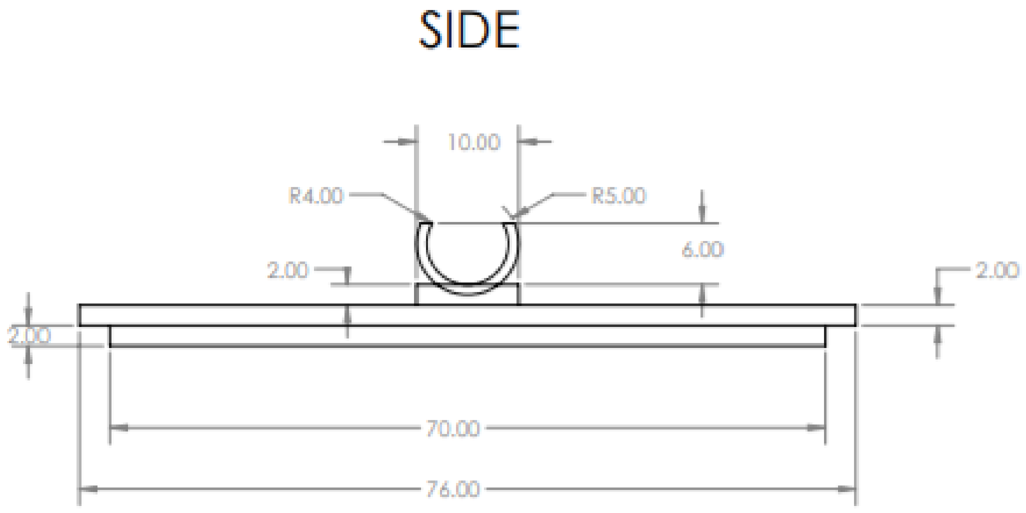


Figure 15: Side of enclosure lid

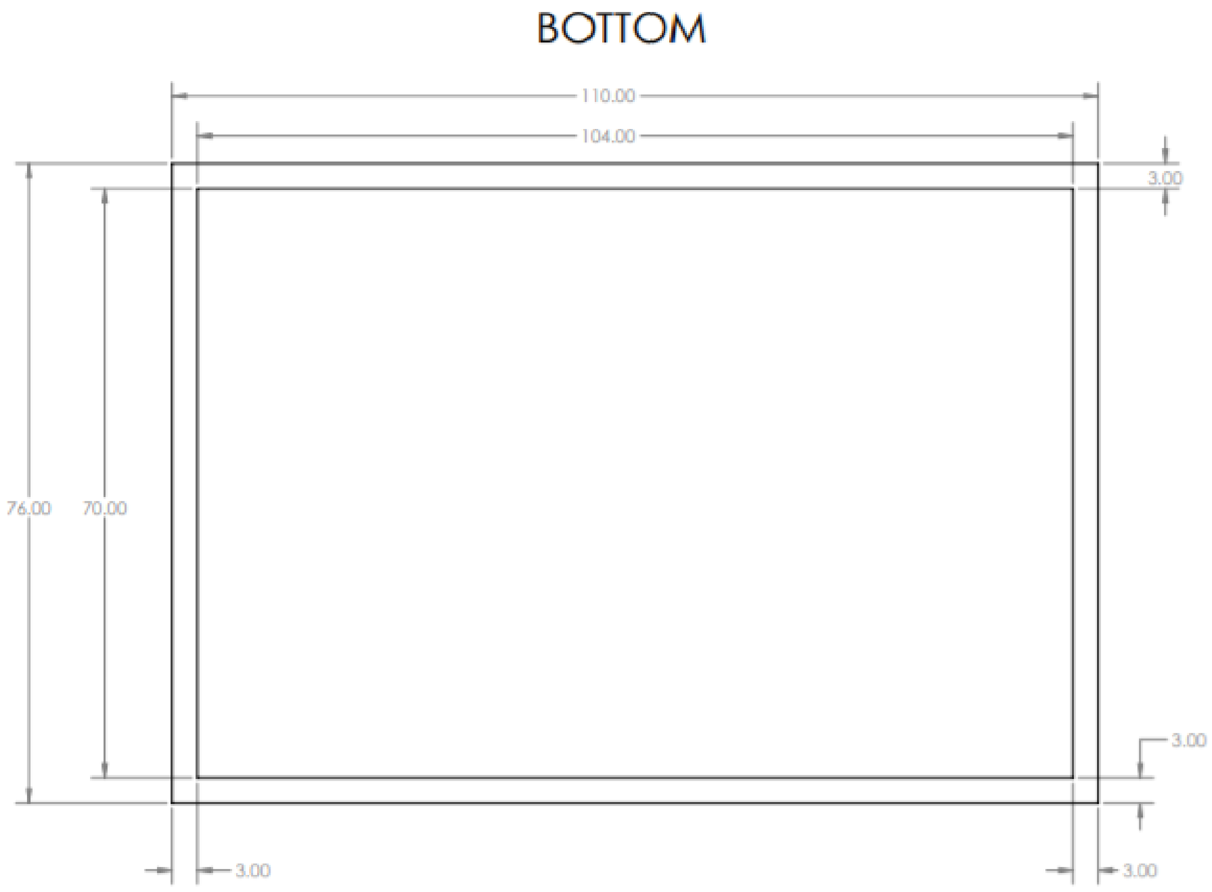


Figure 16: Bottom of enclosure lid

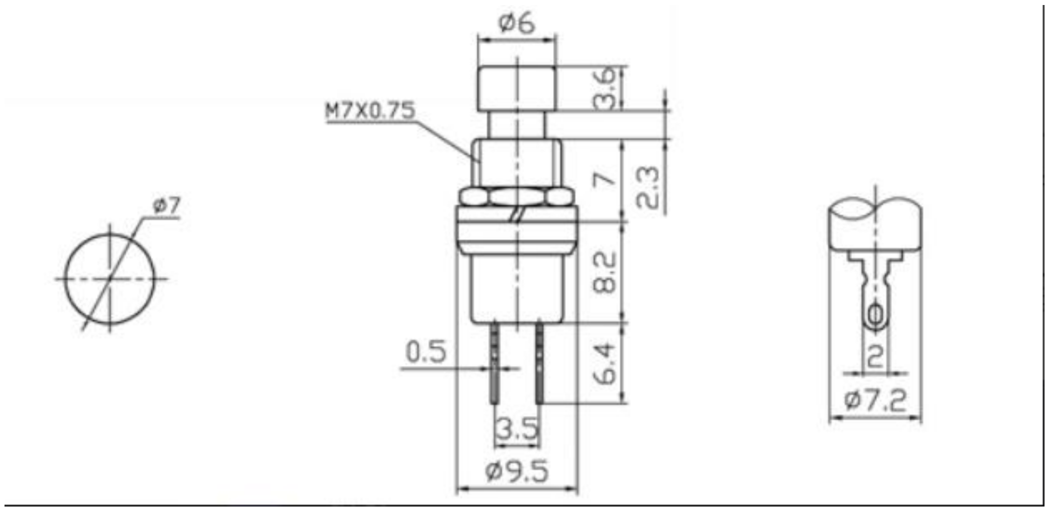


Figure 17: Buttons measurements (from datasheet)

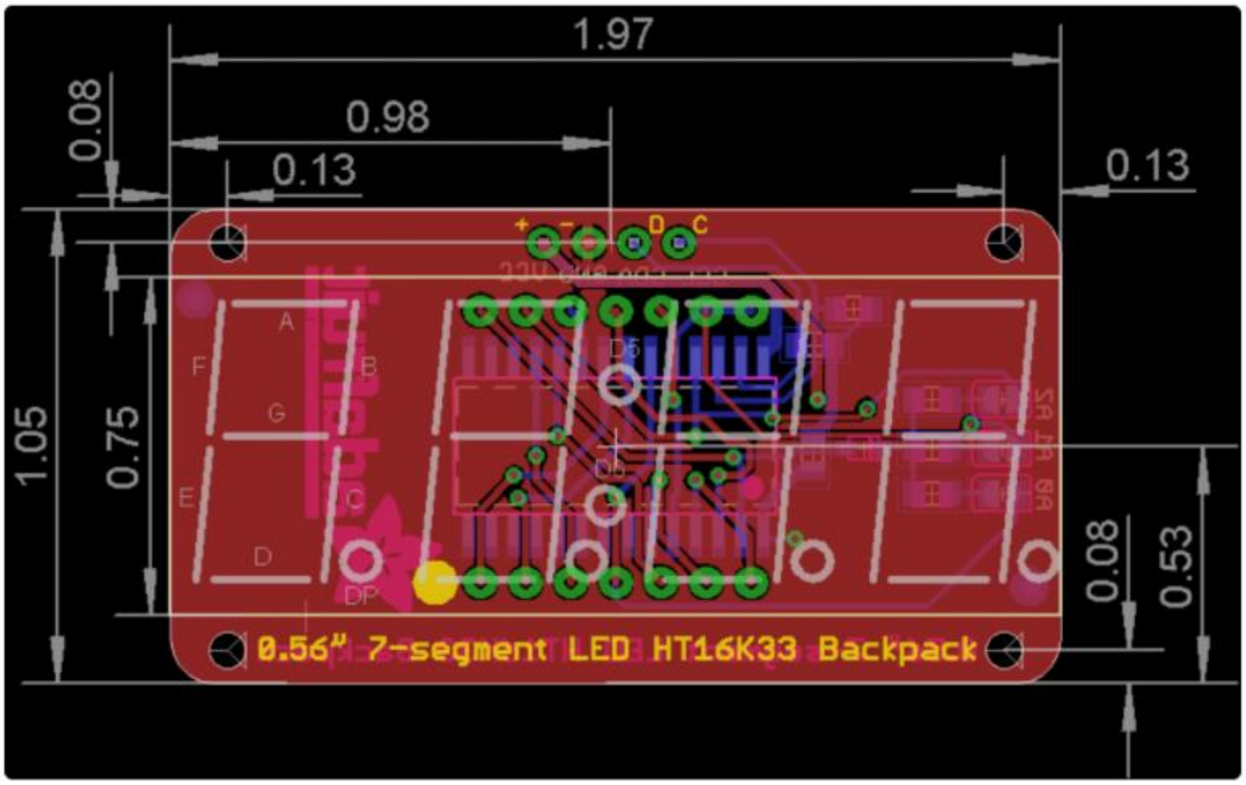


Figure 18: Seven segment display measurements (Provided by Adafruit) (Numbers in inches)

PCB Design

For this project, a theoretical PCB was designed but not implemented. This design could be used in a future version of the project. It only requires one layer so it would be a cheap and effective way of further organizing wires.

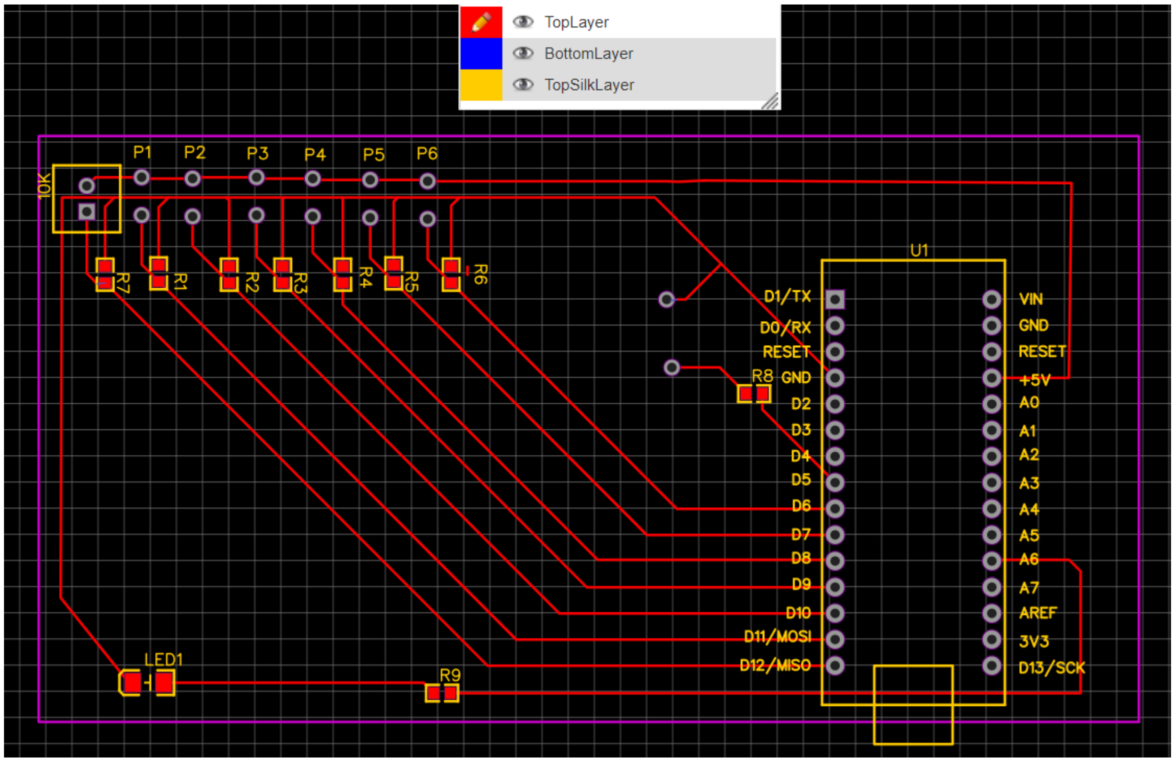


Figure 19: PCB traces and theoretical design for future implementation

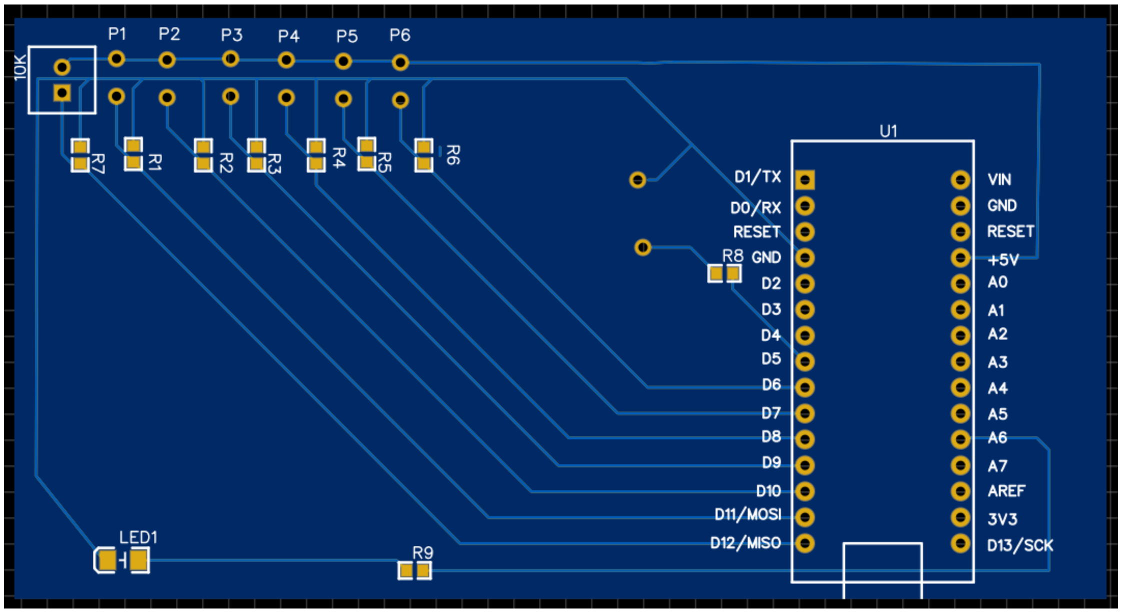
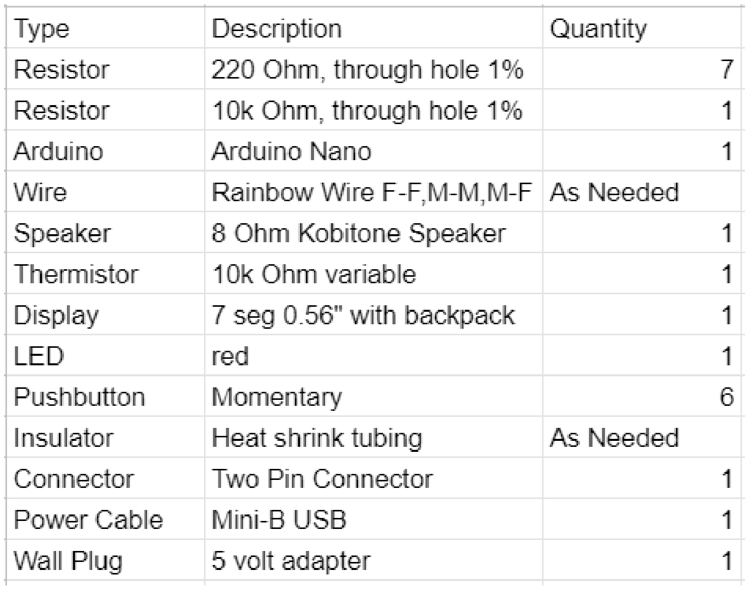


Figure 20: PCB physical appearance if it were to be printed

Parts List

General information for the parts used has been included in Figure 20. An easily accessible Bill of Materials with links to data sheets and information can be accessed here in a google sheets document: https://docs.google.com/spreadsheets/d/1xOXz4FDksTZxfcxoncz6Xkp8biL7jY1Q-sBzcghtst0/edit?usp=sharing

The parts used in this project were all inexpensive and simple. The blocks each had one piece driving each one such as the seven segment display or the speaker but otherwise were composed of simple wires and resistors.



Type	Description	Quantity
Resistor	220 Ohm, through hole 1%	7
Resistor	10k Ohm, through hole 1%	1
Arduino	Arduino Nano	1
Wire	Rainbow Wire F-F,M-M,M-F	As Needed
Speaker	8 Ohm Kobitone Speaker	1
Thermistor	10k Ohm variable	1
Display	7 seg 0.56" with backpack	1
LED	red	1
Pushbutton	Momentary	6
Insulator	Heat shrink tubing	As Needed
Connector	Two Pin Connector	1
Power Cable	Mini-B USB	1
Wall Plug	5 volt adapter	1

Figure 20: Bill of Materials without links

Code

Below the code for the Arduino is included. It is thoroughly commented but has a major part that repeats multiple times at the end. The code is much simpler than its length would suggest because of this, and it could probably be optimized in future models. This code is written for Arduino sketch. This is the last artifact included in the document for ease of viewing of the other artifacts. In the this code could be improved with use of interrupts and functions as right now it uses polling methods that work reasonably well but could be improved.

```
#include <Adafruit_GFX.h>

#include "Adafruit_LEDBackpack.h"

#include <Wire.h>


#define speaker_pin 5 //defines "speaker_pin" for the digital output pin 5

const int therminput = A6; //ADC of the thermistor
const int resetbutton = 9; //6 pushbutton inputs mapped to pins
const int oneminutebutton = 6;
const int fiveminutebutton = 7;
const int startbutton = 8;
const int brightnessbutton = 10;
const int boilerbutton = 11;
const int boilerLED = 12; //LED used to say the boiler is on

int tempADC; //This will record the temperature voltage

int boilerstate = 0; //Initial values are set to 0
int startstate = 0; //This is done to set the system
int oneminutestate = 0; //to its initial stopped state
int fiveminutestate = 0; //These six represent button presses
int resetstate = 0;
int brightnessstate = 0;

int time = 0; //The timer starts with 0 seconds on it
int boileron = 0; //This tracks if the boiler is on (1) or off (0)
int started = 0; //This controls if the system is currently counting down
int brightlevel = 7; //This holds the next brightness level (Medium at start)
int jp1 = 0;  //These four variables help prevent double button presses
int jp5 = 0;
int jpbri = 0;
int jpboil = 0;

Adafruit_7segment matrix = Adafruit_7segment(); //Sets up the matrix for the seven seg

void setup(){
  #ifndef __AVR_ATtiny85__ //Start of seven seg setup
  Serial.begin(9600);
  Serial.println("7 Segment Backpack Test");
#endif
  matrix.begin(0x70); //end of seven seg setup

  pinMode(resetbutton, INPUT); //Sets the 6 buttons as inputs
  pinMode(oneminutebutton, INPUT);
  pinMode(fiveminutebutton, INPUT);
  pinMode(startbutton, INPUT);
  pinMode(brightnessbutton, INPUT);
  pinMode(boilerbutton, INPUT);
  pinMode(boilerLED, OUTPUT); //Sets the boilerLED as an output
}


void loop() { //Main program
  tempADC = analogRead(therminput); //All inputs are read
  boilerstate = digitalRead(boilerbutton);
  brightnessstate = digitalRead(brightnessbutton);
  oneminutestate = digitalRead(oneminutebutton);
  fiveminutestate = digitalRead(fiveminutebutton);
  resetstate = digitalRead(resetbutton);
  startstate = digitalRead(startbutton);

  if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
    time = time + 100; //Adds one minute
    noTone(speaker_pin); //Disables speaker
    jp1 = 2;
  }

  if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
    time = time + 500; //Adds 5 minutes
    noTone(speaker_pin); //Disables speaker
    jp5 = 2;
  }

  if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
    matrix.setBrightness(brightlevel); //Each of these if statements is used
    if(brightlevel == 7){             //To switch to the next brightness level
      brightlevel = 0;
    }
    else if(brightlevel == 0){
      brightlevel = 15;
    }
    else if(brightlevel == 15){
      brightlevel = 7;
    }
    jpbri = 2;
  }

  if(startstate == HIGH){ //If start button is pressed
    started = 1; //Labels the system as starting
  }

  if(resetstate == HIGH){ //If the reset button is pressed
    started = 0; //System is no longer running
    time = 0; //Time is set to 0
    noTone(speaker_pin); //Speaker is turned off
  }

  if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
    if(boileron == 0){ //If boiler is off
      boileron = 1; //Boiler is marked as on
      digitalWrite(boilerLED, HIGH); //LED is turned on
    }
    else{
      boileron = 0; //Boiler is marked as off
      digitalWrite(boilerLED, LOW); //LED is turned off
    }
    jpboil = 2;
  }

  //This section decrements just pressed variables
  //These variables are set to 2 when a button is pressed and are
  //Put to 0 after 1 additional cycle this leaves a gap of .2 seconds
  //Where a button cannot be pressed twice on accident
  if(jpboil != 0){
    jpboil = jpboil - 1;
  }
  if(jpbri != 0){
    jpbri = jpbri - 1;
  }
  if(jp1 != 0){
    jp1 = jp1 - 1;
```

```
 }
 if(jp5 != 0){
   jp5 = jp5 - 1;
 }

if(time % 100 == 0 and time != 0 and started == 1){
   time = time - 40; //At increments of 100 the time is switched to 60 for a minute
 }

 if(time != 0 and started == 1){
 time = time - 1; //Counts down one second
 }
 matrix.print(time, DEC); //Prints to the seven segment the time
 matrix.drawColon(true); //draws a colon
 matrix.writeDisplay(); //writes to the display



 if((tempADC > 968 and boileron == 1) or (time == 0 and started == 1)){
 tone(speaker_pin, 440); //Sets pin 5 to a 440 Hz square wave
 }



 delay(100); //Delays 1/10 of a second
 //At this point taking inputs and doing button related activities
 //are repeated 9 more times before reaching this original part of the loop
 //where time related things are taken care of. This is done
 //to allow appropriate timing for button inputs to not be missed or doubled.
 tempADC = analogRead(therminput); //All inputs are read
 boilerstate = digitalRead(boilerbutton);
 brightnessstate = digitalRead(brightnessbutton);
 oneminutestate = digitalRead(oneminutebutton);
 fiveminutestate = digitalRead(fiveminutebutton);
 resetstate = digitalRead(resetbutton);
 startstate = digitalRead(startbutton);

 if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
   time = time + 100; //Adds one minute
```

```
  noTone(speaker_pin); //Disables speaker
  jp1 = 2;
}

if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
  time = time + 500; //Adds 5 minutes
  noTone(speaker_pin); //Disables speaker
  jp5 = 2;
}

if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
  matrix.setBrightness(brightlevel); //Each of these if statements is used
  if(brightlevel == 7){             //To switch to the next brightness level
    brightlevel = 0;
  }
  else if(brightlevel == 0){
    brightlevel = 15;
  }
  else if(brightlevel == 15){
    brightlevel = 7;
  }
  jpbri = 2;
}

if(startstate == HIGH){ //If start button is pressed
  started = 1; //Labels the system as starting
}

if(resetstate == HIGH){ //If the reset button is pressed
  started = 0; //System is no longer running
  time = 0; //Time is set to 0
  noTone(speaker_pin); //Speaker is turned off
}

if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
  if(boileron == 0){ //If boiler is off
    boileron = 1; //Boiler is marked as on
    digitalWrite(boilerLED, HIGH); //LED is turned on
  }
  else{
    boileron = 0; //Boiler is marked as off
    digitalWrite(boilerLED, LOW); //LED is turned off
  }
  jpboil = 2;
}

if(jpboil != 0){
  jpboil = jpboil - 1;
}
if(jpbri != 0){
  jpbri = jpbri - 1;
}
if(jp1 != 0){
  jp1 = jp1 - 1;
}
if(jp5 != 0){
  jp5 = jp5 - 1;
}
delay(100);//Delays 1/10 of a second
  tempADC = analogRead(therminput); //All inputs are read
boilerstate = digitalRead(boilerbutton);
brightnessstate = digitalRead(brightnessbutton);
oneminutestate = digitalRead(oneminutebutton);
fiveminutestate = digitalRead(fiveminutebutton);
resetstate = digitalRead(resetbutton);
startstate = digitalRead(startbutton);

if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
  time = time + 100; //Adds one minute
  noTone(speaker_pin); //Disables speaker
  jp1 = 2;
}

if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
  time = time + 500; //Adds 5 minutes
  noTone(speaker_pin); //Disables speaker
  jp5 = 2;
}

if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
  matrix.setBrightness(brightlevel); //Each of these if statements is used
  if(brightlevel == 7){             //To switch to the next brightness level
    brightlevel = 0;
  }
  else if(brightlevel == 0){
    brightlevel = 15;
  }
  else if(brightlevel == 15){
    brightlevel = 7;
  }
  jpbri = 2;
}

if(startstate == HIGH){ //If start button is pressed
  started = 1; //Labels the system as starting
}

if(resetstate == HIGH){ //If the reset button is pressed
  started = 0; //System is no longer running
  time = 0; //Time is set to 0
  noTone(speaker_pin); //Speaker is turned off
}

if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
  if(boileron == 0){ //If boiler is off
    boileron = 1; //Boiler is marked as on
    digitalWrite(boilerLED, HIGH); //LED is turned on
  }
  else{
    boileron = 0; //Boiler is marked as off
    digitalWrite(boilerLED, LOW); //LED is turned off
  }
  jpboil = 2;
}
```

```
if(jpboil != 0){
  jpboil = jpboil - 1;
}
if(jpbri != 0){
  jpbri = jpbri - 1;
}
if(jp1 != 0){
  jp1 = jp1 - 1;
}
if(jp5 != 0){
  jp5 = jp5 - 1;
}
delay(100);//Delays 1/10 of a second
  tempADC = analogRead(therminput); //All inputs are read
boilerstate = digitalRead(boilerbutton);
brightnessstate = digitalRead(brightnessbutton);
oneminutestate = digitalRead(oneminutebutton);
fiveminutestate = digitalRead(fiveminutebutton);
resetstate = digitalRead(resetbutton);
startstate = digitalRead(startbutton);

if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
  time = time + 100; //Adds one minute
  noTone(speaker_pin); //Disables speaker
  jp1 = 2;
}

if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
  time = time + 500; //Adds 5 minutes
  noTone(speaker_pin); //Disables speaker
  jp5 = 2;
}

if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
  matrix.setBrightness(brightlevel); //Each of these if statements is used
  if(brightlevel == 7){             //To switch to the next brightness level
    brightlevel = 0;
  }
  else if(brightlevel == 0){
    brightlevel = 15;
  }
  else if(brightlevel == 15){
    brightlevel = 7;
  }
  jpbri = 2;
}

if(startstate == HIGH){ //If start button is pressed
  started = 1; //Labels the system as starting
}

if(resetstate == HIGH){ //If the reset button is pressed
  started = 0; //System is no longer running
  time = 0; //Time is set to 0
  noTone(speaker_pin); //Speaker is turned off
}

if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
  if(boileron == 0){ //If boiler is off
    boileron = 1; //Boiler is marked as on
    digitalWrite(boilerLED, HIGH); //LED is turned on
  }
  else{
    boileron = 0; //Boiler is marked as off
    digitalWrite(boilerLED, LOW); //LED is turned off
  }
  jpboil = 2;
}

if(jpboil != 0){
  jpboil = jpboil - 1;
}
if(jpbri != 0){
  jpbri = jpbri - 1;
}
```

```
if(jp1 != 0){
  jp1 = jp1 - 1;
}
if(jp5 != 0){
  jp5 = jp5 - 1;
}
delay(100);//Delays 1/10 of a second
  tempADC = analogRead(therminput); //All inputs are read
boilerstate = digitalRead(boilerbutton);
brightnessstate = digitalRead(brightnessbutton);
oneminutestate = digitalRead(oneminutebutton);
fiveminutestate = digitalRead(fiveminutebutton);
resetstate = digitalRead(resetbutton);
startstate = digitalRead(startbutton);

if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
  time = time + 100; //Adds one minute
  noTone(speaker_pin); //Disables speaker
  jp1 = 2;
}

if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
  time = time + 500; //Adds 5 minutes
  noTone(speaker_pin); //Disables speaker
  jp5 = 2;
}

if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
  matrix.setBrightness(brightlevel); //Each of these if statements is used
  if(brightlevel == 7){              //To switch to the next brightness level
    brightlevel = 0;
  }
  else if(brightlevel == 0){
    brightlevel = 15;
  }
  else if(brightlevel == 15){
    brightlevel = 7;
  }
  jpbri = 2;
}

if(startstate == HIGH){ //If start button is pressed
  started = 1; //Labels the system as starting
}

if(resetstate == HIGH){ //If the reset button is pressed
  started = 0; //System is no longer running
  time = 0; //Time is set to 0
  noTone(speaker_pin); //Speaker is turned off
}

if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
  if(boileron == 0){ //If boiler is off
    boileron = 1; //Boiler is marked as on
    digitalWrite(boilerLED, HIGH); //LED is turned on
  }
  else{
    boileron = 0; //Boiler is marked as off
    digitalWrite(boilerLED, LOW); //LED is turned off
  }
  jpboil = 2;
}

if(jpboil != 0){
  jpboil = jpboil - 1;
}
if(jpbri != 0){
  jpbri = jpbri - 1;
}
if(jp1 != 0){
  jp1 = jp1 - 1;
}
if(jp5 != 0){
  jp5 = jp5 - 1;
}
delay(100);//Delays 1/10 of a second
```

```
  tempADC = analogRead(therminput); //All inputs are read
boilerstate = digitalRead(boilerbutton);
brightnessstate = digitalRead(brightnessbutton);
oneminutestate = digitalRead(oneminutebutton);
fiveminutestate = digitalRead(fiveminutebutton);
resetstate = digitalRead(resetbutton);
startstate = digitalRead(startbutton);

if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
  time = time + 100; //Adds one minute
  noTone(speaker_pin); //Disables speaker
  jp1 = 2;
}

if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
  time = time + 500; //Adds 5 minutes
  noTone(speaker_pin); //Disables speaker
  jp5 = 2;
}

if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
  matrix.setBrightness(brightlevel); //Each of these if statements is used
  if(brightlevel == 7){             //To switch to the next brightness level
    brightlevel = 0;
  }
  else if(brightlevel == 0){
    brightlevel = 15;
  }
  else if(brightlevel == 15){
    brightlevel = 7;
  }
  jpbri = 2;
}

if(startstate == HIGH){ //If start button is pressed
  started = 1; //Labels the system as starting
}
```

```
if(resetstate == HIGH){ //If the reset button is pressed
  started = 0; //System is no longer running
  time = 0; //Time is set to 0
  noTone(speaker_pin); //Speaker is turned off
}

if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
  if(boileron == 0){ //If boiler is off
    boileron = 1; //Boiler is marked as on
    digitalWrite(boilerLED, HIGH); //LED is turned on
  }
  else{
    boileron = 0; //Boiler is marked as off
    digitalWrite(boilerLED, LOW); //LED is turned off
  }
  jpboil = 2;
}

if(jpboil != 0){
  jpboil = jpboil - 1;
}
if(jpbri != 0){
  jpbri = jpbri - 1;
}
if(jp1 != 0){
  jp1 = jp1 - 1;
}
if(jp5 != 0){
  jp5 = jp5 - 1;
}
delay(100);//Delays 1/10 of a second
  tempADC = analogRead(therminput); //All inputs are read
boilerstate = digitalRead(boilerbutton);
brightnessstate = digitalRead(brightnessbutton);
oneminutestate = digitalRead(oneminutebutton);
fiveminutestate = digitalRead(fiveminutebutton);
resetstate = digitalRead(resetbutton);
startstate = digitalRead(startbutton);
```

```
if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
  time = time + 100; //Adds one minute
  noTone(speaker_pin); //Disables speaker
  jp1 = 2;
}

if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
  time = time + 500; //Adds 5 minutes
  noTone(speaker_pin); //Disables speaker
  jp5 = 2;
}

if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
  matrix.setBrightness(brightlevel); //Each of these if statements is used
  if(brightlevel == 7){             //To switch to the next brightness level
    brightlevel = 0;
  }
  else if(brightlevel == 0){
    brightlevel = 15;
  }
  else if(brightlevel == 15){
    brightlevel = 7;
  }
  jpbri = 2;
}

if(startstate == HIGH){ //If start button is pressed
  started = 1; //Labels the system as starting
}

if(resetstate == HIGH){ //If the reset button is pressed
  started = 0; //System is no longer running
  time = 0; //Time is set to 0
  noTone(speaker_pin); //Speaker is turned off
}

if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
```

```
  if(boileron == 0){ //If boiler is off
    boileron = 1; //Boiler is marked as on
    digitalWrite(boilerLED, HIGH); //LED is turned on
  }
  else{
    boileron = 0; //Boiler is marked as off
    digitalWrite(boilerLED, LOW); //LED is turned off
  }
  jpboil = 2;
}

if(jpboil != 0){
  jpboil = jpboil - 1;
}
if(jpbri != 0){
  jpbri = jpbri - 1;
}
if(jp1 != 0){
  jp1 = jp1 - 1;
}
if(jp5 != 0){
  jp5 = jp5 - 1;
}
delay(100);//Delays 1/10 of a second
  tempADC = analogRead(therminput); //All inputs are read
boilerstate = digitalRead(boilerbutton);
brightnessstate = digitalRead(brightnessbutton);
oneminutestate = digitalRead(oneminutebutton);
fiveminutestate = digitalRead(fiveminutebutton);
resetstate = digitalRead(resetbutton);
startstate = digitalRead(startbutton);

if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
  time = time + 100; //Adds one minute
  noTone(speaker_pin); //Disables speaker
  jp1 = 2;
}
```

```
if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
  time = time + 500; //Adds 5 minutes
  noTone(speaker_pin); //Disables speaker
  jp5 = 2;
}

if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
  matrix.setBrightness(brightlevel); //Each of these if statements is used
  if(brightlevel == 7){             //To switch to the next brightness level
    brightlevel = 0;
  }
  else if(brightlevel == 0){
    brightlevel = 15;
  }
  else if(brightlevel == 15){
    brightlevel = 7;
  }
  jpbri = 2;
}

if(startstate == HIGH){ //If start button is pressed
  started = 1; //Labels the system as starting
}

if(resetstate == HIGH){ //If the reset button is pressed
  started = 0; //System is no longer running
  time = 0; //Time is set to 0
  noTone(speaker_pin); //Speaker is turned off
}

if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
  if(boileron == 0){ //If boiler is off
    boileron = 1; //Boiler is marked as on
    digitalWrite(boilerLED, HIGH); //LED is turned on
  }
  else{
    boileron = 0; //Boiler is marked as off
    digitalWrite(boilerLED, LOW); //LED is turned off
  }
  jpboil = 2;
}

if(jpboil != 0){
  jpboil = jpboil - 1;
}
if(jpbri != 0){
  jpbri = jpbri - 1;
}
if(jp1 != 0){
  jp1 = jp1 - 1;
}
if(jp5 != 0){
  jp5 = jp5 - 1;
}
delay(100);//Delays 1/10 of a second
  tempADC = analogRead(therminput); //All inputs are read
boilerstate = digitalRead(boilerbutton);
brightnessstate = digitalRead(brightnessbutton);
oneminutestate = digitalRead(oneminutebutton);
fiveminutestate = digitalRead(fiveminutebutton);
resetstate = digitalRead(resetbutton);
startstate = digitalRead(startbutton);

if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
  time = time + 100; //Adds one minute
  noTone(speaker_pin); //Disables speaker
  jp1 = 2;
}

if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
  time = time + 500; //Adds 5 minutes
  noTone(speaker_pin); //Disables speaker
  jp5 = 2;
}

if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
```

```
  matrix.setBrightness(brightlevel); //Each of these if statements is used
  if(brightlevel == 7){             //To switch to the next brightness level
    brightlevel = 0;
  }
  else if(brightlevel == 0){
    brightlevel = 15;
  }
  else if(brightlevel == 15){
    brightlevel = 7;
  }
  jpbri = 2;
}

if(startstate == HIGH){ //If start button is pressed
  started = 1; //Labels the system as starting
}

if(resetstate == HIGH){ //If the reset button is pressed
  started = 0; //System is no longer running
  time = 0; //Time is set to 0
  noTone(speaker_pin); //Speaker is turned off
}

if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
  if(boileron == 0){ //If boiler is off
    boileron = 1; //Boiler is marked as on
    digitalWrite(boilerLED, HIGH); //LED is turned on
  }
  else{
    boileron = 0; //Boiler is marked as off
    digitalWrite(boilerLED, LOW); //LED is turned off
  }
  jpboil = 2;
}

if(jpboil != 0){
  jpboil = jpboil - 1;
}
```

```
if(jpbri != 0){
  jpbri = jpbri - 1;
}
if(jp1 != 0){
  jp1 = jp1 - 1;
}
if(jp5 != 0){
  jp5 = jp5 - 1;
}
delay(100);//Delays 1/10 of a second
  tempADC = analogRead(therminput); //All inputs are read
boilerstate = digitalRead(boilerbutton);
brightnessstate = digitalRead(brightnessbutton);
oneminutestate = digitalRead(oneminutebutton);
fiveminutestate = digitalRead(fiveminutebutton);
resetstate = digitalRead(resetbutton);
startstate = digitalRead(startbutton);

if(oneminutestate == HIGH and jp1 == 0){ //If one minute button is pressed
  time = time + 100; //Adds one minute
  noTone(speaker_pin); //Disables speaker
  jp1 = 2;
}

if(fiveminutestate == HIGH and jp5 == 0){ //If five minute button is pressed
  time = time + 500; //Adds 5 minutes
  noTone(speaker_pin); //Disables speaker
  jp5 = 2;
}

if(brightnessstate == HIGH and jpbri == 0){ //If brightness button is pressed
  matrix.setBrightness(brightlevel); //Each of these if statements is used
  if(brightlevel == 7){             //To switch to the next brightness level
    brightlevel = 0;
  }
  else if(brightlevel == 0){
    brightlevel = 15;
  }
  else if(brightlevel == 15){
    brightlevel = 7;
  }
  jpbri = 2;
}

if(startstate == HIGH){ //If start button is pressed
  started = 1; //Labels the system as starting
}

if(resetstate == HIGH){ //If the reset button is pressed
  started = 0; //System is no longer running
  time = 0; //Time is set to 0
  noTone(speaker_pin); //Speaker is turned off
}

if(boilerstate == HIGH and jpboil == 0){ //If boiler button is pressed
  if(boileron == 0){ //If boiler is off
    boileron = 1; //Boiler is marked as on
    digitalWrite(boilerLED, HIGH); //LED is turned on
  }
  else{
    boileron = 0; //Boiler is marked as off
    digitalWrite(boilerLED, LOW); //LED is turned off
  }
  jpboil = 2;
}

if(jpboil != 0){
  jpboil = jpboil - 1;
}
if(jpbri != 0){
  jpbri = jpbri - 1;
}
if(jp1 != 0){
  jp1 = jp1 - 1;
}
if(jp5 != 0){
    jp5 = jp5 - 1;

  }

  delay(100);//Delays 1/10 of a second

}
```